CS 142 Midterm 2

Version 0.9

Instructor: R. P. Burton

Monday, March 11, 2013 thru Thursday, March 14, 2013, 6:00 p.m.

Late penalty is 20 points per day, weekend days included, advancing at 6:01 p.m. each day

Allowed materials include your CS 142 course text, your own notes, your own prior lab solutions, the CS 142 course website, and cplusplus.com

Disallowed materials include all other text resources, all other Internet

resources, and all neighbors (remember, everyone is thy neighbor)

**Instructions**

1. This midterm consists of a C++ programming problem. Read and understand the statement of the problem completely before beginning to design, code, and test. Understanding the problem correctly is part of the examination. If something seems unclear, ask a CS 142 TA (but no one else) for clarification.

2. As part of your design, consider test cases that will establish the correctness of your solution.

3. Produce a solution, which consists of *your* C++ code, with a comment at the beginning of your file which includes your name, your student ID number, and “CS 142 Winter 2013 Midterm 2.” Save your complete source file(s) using the online submission script. Attribute any code taken from or based on other sources (except for the course text and the course website). Code taken from or based on other sources is worth half credit.

4. You may pose questions to the CS 142 TAs at any time. However, the TAs generally are not permitted to answer questions related to design, C++ implementation, debugging, or testing.

5. Code well. This includes choosing good names for identifiers, avoiding magic numbers, choosing good control structures, decomposing into meaningful functions, formatting in a communicative way, etc. so that your code is reasonably self-documenting. Provide comments where required or appropriate, especially to help the TAs understand what your code is intended to do. Remember, you will be graded not only on whether your code (when entered, compiled, and executed) accomplishes the specified task, but also on how clearly and efficiently it accomplishes the specified task, and on how closely you adhere to the stated requirements and to good programming practices.

6. When you are finished, score your solution as indicated on the Scoring Sheet. fabs (your\_score – the\_TAs\_score) \* 0.5 – 5.0 will be subtracted from your score. Then go to the course website and follow the link labeled “Submit Exam” in the Exam Menu.

7. Sign here to request that your midterm be graded and to certify that no unfair information related to the midterm has been received by you, either directly or indirectly, and that none has been or will be conveyed by you. If we discover that you cheated or assisted someone in cheating, intentionally or unintentionally (including accidentally), your score for this exam may (and probably will) be rand() % 1. Discipline also may include Honor Code Office involvement, which can lead to loss of opportunities at Brigham Young University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Name)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, 2013

(Signature) (Date)

**Mumblety Jumble**

**Requirements**

Design, implement and test a C++ program that (a) prompts the user for the names of an input file and an output file, (b) reads the input file, (c) “jumbles” each word in the file, and (d) creates the output file consisting of the “jumbled” words in their original order.

Input

The input file is built by composing a syntactically correct and semantically meaningful memo (being careful not to disclose anything you would prefer to keep private). You may write the memo using any language in which you are fluent provided it uses the Latin alphabet (no worries, mate; that includes English, Spanish, Portuguese, Swedish, Crimean Tartar, and dozens of other languages). Use punctuation marks common to your selected language such as periods, commas, semicolons, colons, exclamation marks, single and double quotation marks, apostrophes, and question marks. Save the text of the memo as a “.txt” file.

Jumbling

A word is “jumbled” by preserving its first and last letters, and by rearranging any letters between the first and last letters so that these letters are in random order. For example, the word “jumble” might become “jmbule” or “jbumle” or “jumlbe”, but the word “the” remains “the”. Any number of punctuation marks preceding or following words retain their positions. Any number of punctuation marks interior to words (such as interior apostrophes) are treated the same as letters interior to words.

Output

The output file consists of the (jumbled) words in the original order and punctuation marks exterior to a word in their original positions. Save the jumbled text as a “.txt” file.

Submission

Print and attach the original memo and the “jumbled” memo to the hardcopy of your midterm.

**Implementation Considerations**

* Your main function should consist only of
  + needed prompts together with features for capturing the user’s response to the prompts;
  + calls to functions for carrying out the reading, jumbling, and writing; and
  + features to permit the user to repeat or terminate the process.

Place all of your prompts and features for capturing user input in the main function.

* Be alert to each of the following:
  + The input file needs to exist.
  + The user may need to be prompted repeatedly until he correctly specifies the input file.
  + The input file should not be edited in any way.
  + The user should select different names for the input file and the output file.
  + If the output file does not exist, it will be created; if it exists, its name will be “stolen” and used for the new output file.
* If you choose a language other than English, avoid syntax conventions that would be incompatible with English or that preclude use of common English punctuation marks ( .,:;!’”’?)
* As an option, you may discard and ignore newline characters in the input file; when creating the output file, simply place a space after each word (and any associated punctuation marks), and place a newline character after every group of 12 words.
* Subsequent executions of your program using the same input file should be expected to produce different output files because the randomization process will be seeded differently.

**Midterm 2 Scoring Sheet**

Name:\_\_\_\_\_\_\_\_\_\_Section #:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Day Submitted: \_\_\_\_\_\_\_\_\_\_\_\_

Days Late: \_\_\_\_\_ x 20= \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Caution**: You will forfeit 10 points if your code has syntax errors and does not compile.

Student Grading TA Grading

**\_\_\_/ 20 pts \_\_\_/ 20 pts – File I/O**

\_\_\_/ 10 pts \_\_\_/10 pts –Accepts only valid input file names and reprompts on bad input

\_\_\_/ 10 pts \_\_\_/ 10 pts –Accepts an output file name and creates the file if it doesn’t exist

**\_\_\_/ 60 pts \_\_\_/ 60 pts – The Art of Jumbling**

\_\_\_/ 15 pts \_\_\_/ 15 pts –Correctly jumbles each word in the original memo

\_\_\_/ 15 pts \_\_\_/ 15 pts –Correctly deals with all punctuation in the original memo

\_\_\_/ 10 pts \_\_\_/ 10 pts –Appropriately randomizes the “jumbling” of each word, including setting the random seed.

\_\_\_/ 10 pts \_\_\_/ 10 pts –Uses functions to implement the program in accordance with the Requirements. This includes the main function.

\_\_\_/ 10 pts \_\_\_/ 10 pts – The two required memos are submitted with the exam packet

**\_\_\_/ 20 pts \_\_\_/ 20 pts – Coding Style**

\_\_\_/ 10 pts \_\_\_/ 10 pts – Correct indentation, visual neatness

\_\_\_/ 10 pts \_\_\_/ 10 pts – Descriptive variable names, comments as needed

**\_\_\_/ 100 pts \_\_\_/ 100 pts – subtotal** (before late penalties)

**\_\_\_/ 100 pts \_\_\_/ 100 pts – subtotal** (after late penalties)

\_\_\_ Instruction 6 adjustment

\_\_\_ **TOTAL**

Student –to-TA Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TA-to-Student Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Graded by:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Grader’s Printed Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_